Modalità di utilizzo del codice

Query

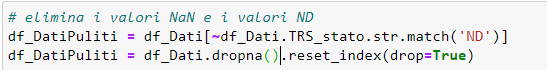
Il codice è stato scritto utilizzando i Notebook Jupyter, integrati con librerie pandas, numpy, matplotlib e excel su linguaggio di programmazione python.

Le query sono state implementate usando Python con l’ausilio della libreria pandas.

Inizialmente è stato creato un dataframe **df\_DatiSensori** direttamente dal file excel ‘**Dati\_gruppo1.xlsx**’ e poi è stata fatto un dataframe copia **df\_Dati** per lavorarci.



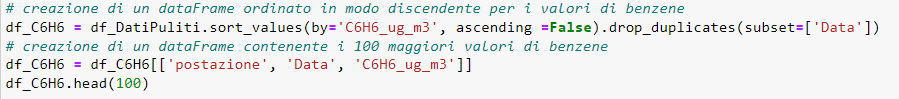
Successivamente effettuiamo una pulizia del dataset appena importato. Innanzitutto andiamo a rinominare le colonne ed effettuiamo operazioni per il riconoscimento di eventuali valori 0 consecutivi che potrebbero significare fallimento dei sensori nell’invio dei dati, dei valori nulli e degli errori riportati con ‘**ND**’ nel file excel. Infine andiamo a effettuare un reset degli indici del dataframe per non avere salti tra un indice ed un altro.



**Le 100 registrazioni con il maggior livello di benzene**

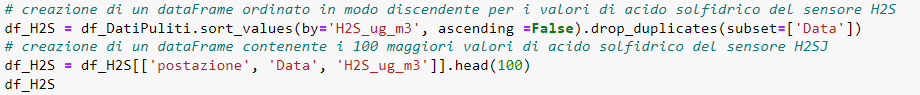
Nella prima query abbiamo creato un dataframe ordinato in modo discendete per i valori di benzene, andando ad eliminare qualsiasi duplicato riguardo alla data.

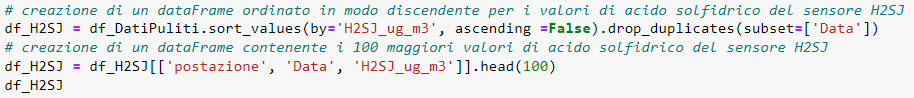
Il dataframe finale conterrà la postazione, la data e il valore del benzene, mostrando i primi cento.



**Le 100 registrazioni con il maggior livello di acido solfidrico per i sensori H2S e H2SJ**

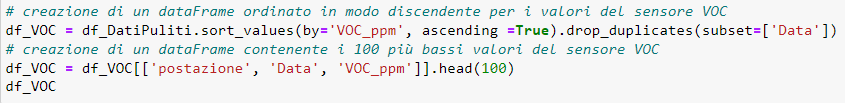
Nella seconda query creiamo due dataframe, **df\_H2S** e **df\_H2SJ**, per i due sensori ed effettuiamo le stesse operazioni effettuate per la prima query.

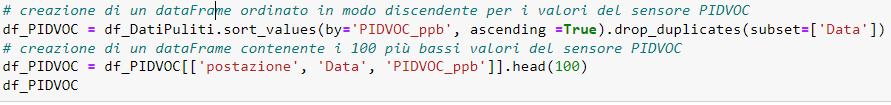




**Le 100 registrazioni con i più bassi livelli di VOC per i sensori VOC e PIDVOC**

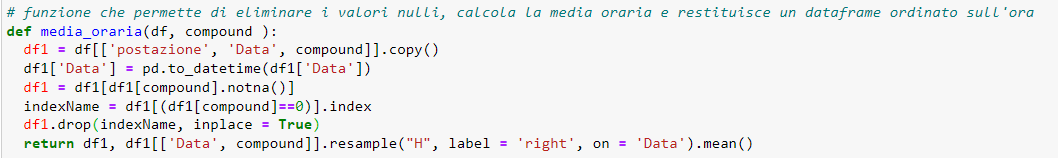
Nella terza query creiamo due dataframe, **df\_VOC** e **df\_PIDVOC**, per i due sensori ed effettuiamo le stesse operazioni effettuate per la prima query, con la differenza che in questo caso andremo ad ordinare i valori in maniera ascendente, cosi da prendere i 100 valori più bassi per i sensori.





**Le 50 ore con il più alto/basso livello medio di benzene**

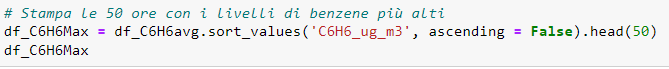
Per questa query è stata creata una funzione che permette di eliminare i valori nulli, calcolare la media oraria e restituire un dataframe ordinato sull’orae che contiene solo le colonne postazione, data e il composto interessato.

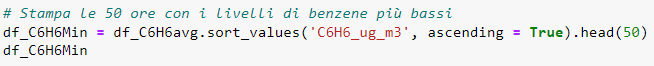


Dopodiché si è pensato di creare un dataframe richiamando la funzione **media\_oraria**, andando ad aggiungere come parametri della funzione il dataframe **df\_DatiPuliti** e il composto interessato, ovvero il benzene.



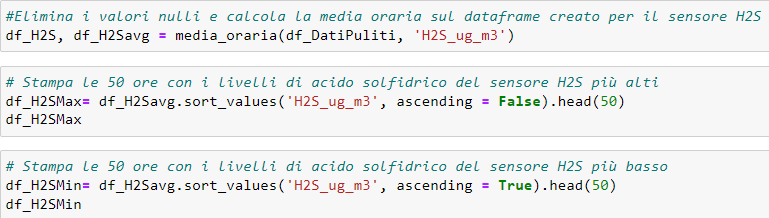
Infine è stata calcolata la più alta e la più bassa media oraria di benzene, andando poi a mostrare solo i primi 50 valori, modificando solo il parametro **ascending** della funzione **sort\_values**.

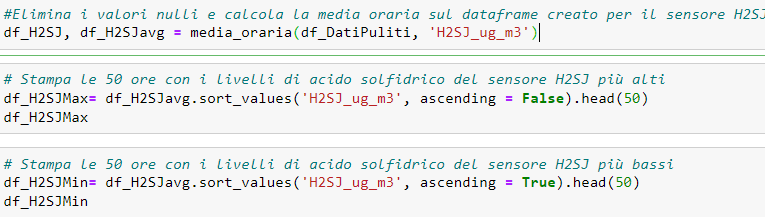




**Le 50 ore con il più alto/basso livello medio di acido solfidrico secondo i sensori H2S e H2SJ**

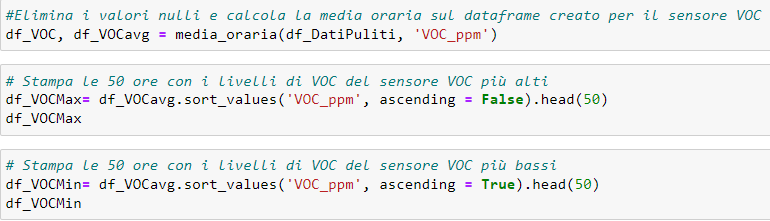
Sono state effettuate le stesse operazioni della query precedente, duplicate per entrambe i sensori **H2S** e **H2SJ**.

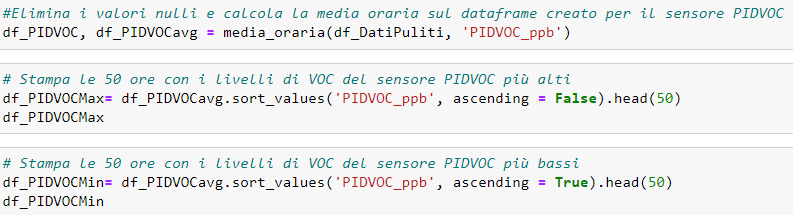




**Le 50 ore con il più alto/basso livello medio di VOC secondo i sensori VOC e PIDVOC**

Sono state effettuate le stesse operazioni della query precedente, duplicate per entrambe i sensori **VOC** e **PIDVOC**.





**Le 3 giornate con il maggior numero di fallimenti nell’invio dei dati**

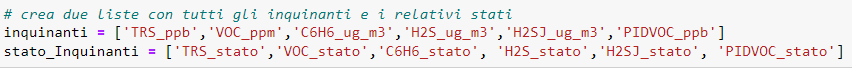
Da fare

**Le 3 giornate con il minor numero di fallimenti nell’invio dei dati**

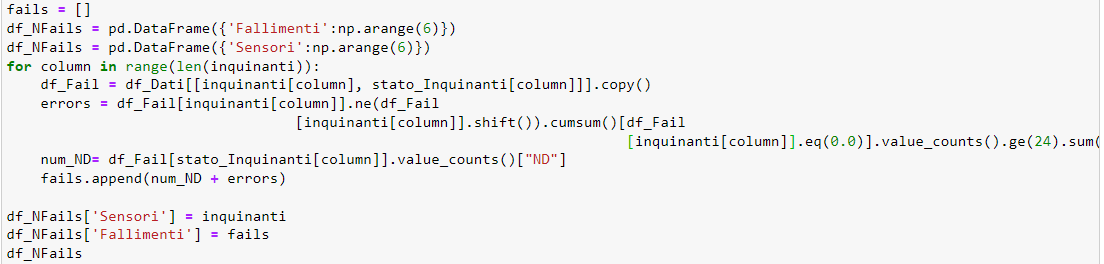
Da fare

**Il numero medio di fallimenti nell’invio per sensore**

Per questa query sono state create due liste, **inquinanti** e **stato\_inquinanti**, da usare nella funzione successiva.



È stato sviluppato un metodo che permette di contare la somma dei fallimenti per ogni sensore andando poi ad inserirli in un nuovo dataframe **df\_NFails**. Innanzitutto creo una lista di fallimenti, **fails**, e poi inizializzo il nuovo dataframe **df\_NFails** a 2 colonne di nome **Sensori** e **Fallimenti**. Vado ad effettuare un ciclo sulla lunghezza di una delle due liste (quale è indifferente, avendo entrambe la stessa lunghezza). Nel ciclo creo un dataframe temporaneo contenente solo le colonne **inquinante** e **stato\_inquinante**. La variabile **errors** restituisce il numero di fallimenti dovuti alla presenza di 0 consecutivi (come fatto per la pulizia del dataset, abbiamo scelto un valore di 24 zeri consecutivi che equivalgono a 2 ore di invii di dati considerati falliti), mentre **num\_ND** restituisce il numero di valori **ND** presenti. Infine questi valori vengono sommati e aggiunti alla lista **fails**. Fuori dal ciclo for viene popolato il dataframe **df\_NFails** con la colonna sensori uguagliata alla lista inquinanti e la colonna fallimenti con **fails**.



Viene aggiunta una colonna ‘**Dati TOT**’ che comprende la lunghezza dei dati per ogni sensore ed infine viene creata un’ultima colonna chiamata ‘**Media**’ contente la media tra la colonna ‘**Fallimenti**’ e ‘**Dati TOT**’.



**Il sensore con il numero massimo di fallimenti**

Per questa query è stato utilizzato lo stesso dataframe creato nella query precedente ed è stata fatta una chiamata **max()** al dataframe **df\_NFails** sulla colonna ‘**Fallimenti**’.



**Il sensore con il numero minimo di fallimenti**

Per questa query è stato utilizzato lo stesso dataframe creato nella query precedente ed è stata fatta una chiamata **min()** al dataframe **df\_NFails** sulla colonna ‘**Fallimenti**’

